Одесский колледж компьютерных технологий «Сервер»

Отчет по практической работе №5

«Полиморфизм (позднее связывание). Восходящее преобразование»

Выполнил:

студент группы К19.1

Фучило Богдан

Проверил:

Рудниченко Н.Д.

Одесса 2022

Содержание

[ВВЕДЕНИЕ 2](#_Toc461126358)

[ТЕОРИТИЧЕСКАЯ ЧАСТЬ 3](#_Toc461126359)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 6](#_Toc461126360)

[ВЫВОД 14](#_Toc461126361)

[ЛИТЕРАТУРА 15](#_Toc461126362)

# ВВЕДЕНИЕ

* Ознакомиться с механизмом полиморфизма в ООП;
* Разобраться со статическим и динамическим связыванием;
* Научиться использовать восходящее преобразование.

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

Полиморфизм является третьей важной особенностью объектно-ориентированных языков, вместе с абстракцией и наследованием.

Он представляет еще одну степень отделения интерфейса от реализации, разъединения ***что*** от ***как***. Полиморфизм улучшает организацию кода и его читаемость, а также способствует созданию **расширяемых** программ, которые могут «расти» не только в процессе начальной разработки проекта, но и при добавлении новых возможностей.

**ЗАДАНИЯ НА ЛАБОРАТОРНУЮ РАБОТУ:**

**Задание 1.**

1. Создайте класс Person, который должен содержать следующие поля и геттеры\сеттеры для этих полей:

* Фамилия;
* Имя;
* Возраст.

Класс Person изображен на рисунке 1

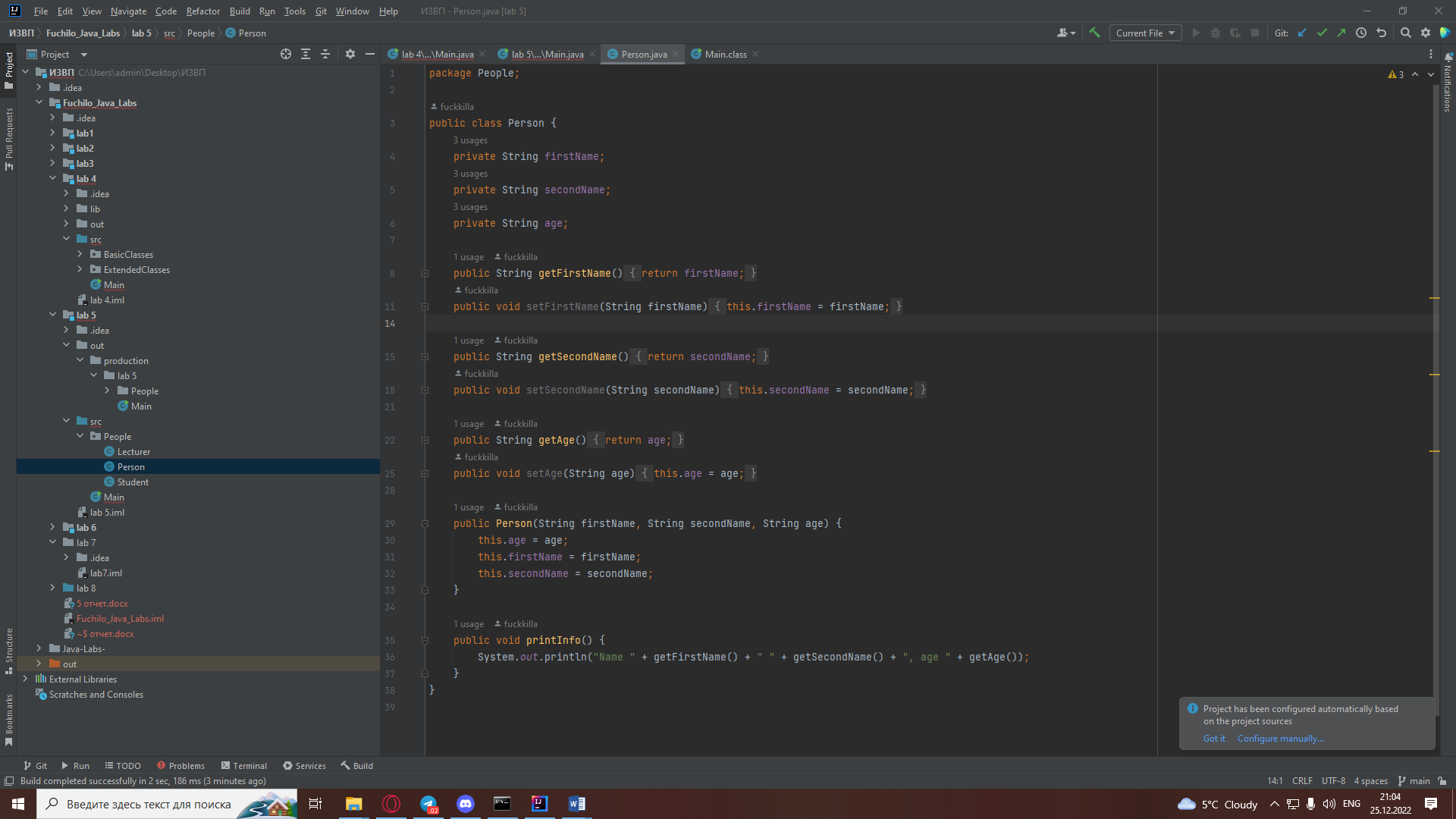


Рисунок 1

Создайте метод printInfo(), который бы возвращал строку следующего формата:

|  |
| --- |
| Человек <фамилия> <имя>, возраст: <возраст> |

Метод printInfo() изображен на рисунке 2

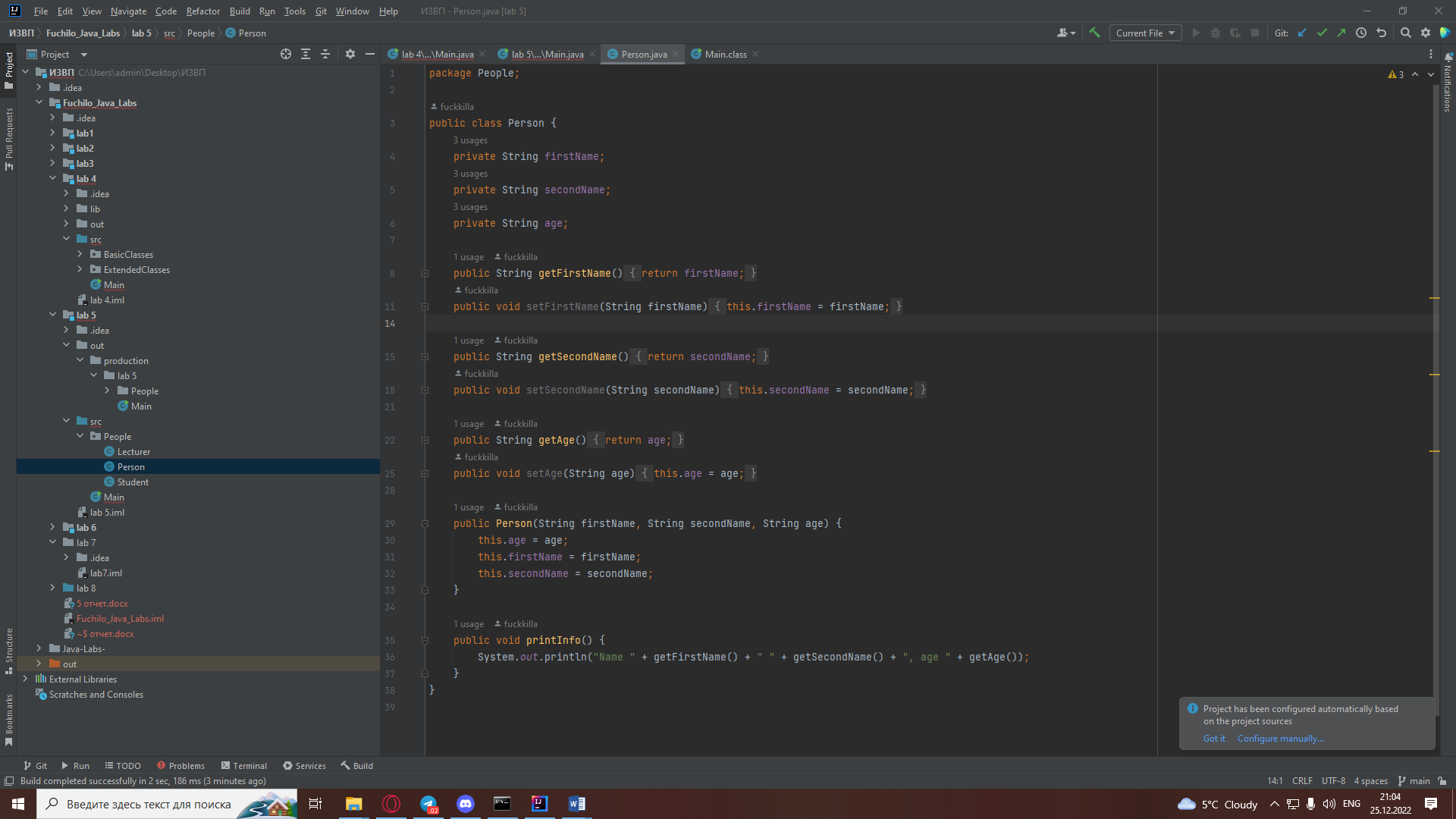


Рисунок 2

2. Создайте класс Student, который должен наследоваться от класса Person. Добавьте дополнительные поля, и геттеры\сеттеры для этих полей:

* Группа;
* Номер студенческого билета.

Класс Student изображен на рисунке 3

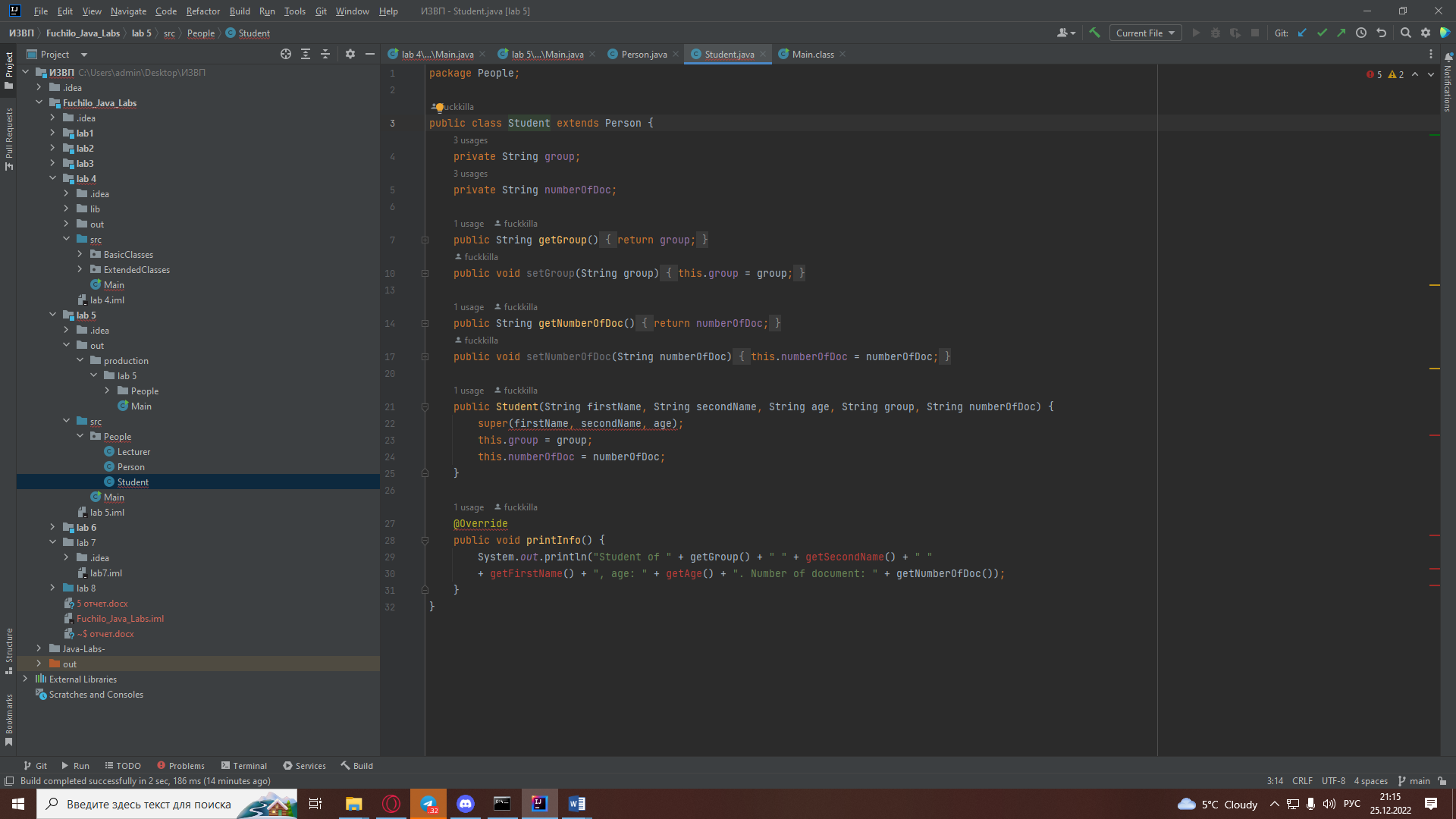


Рисунок 3

Переопределите метод printInfo(), который должен выводить строку следующего формата:

|  |
| --- |
| Студент группы <группа> <фамилия> <имя>, возраст: <возраст>. Номер студенческого билета: <номер> |

Метод printInfo() изображен на рисунке 4

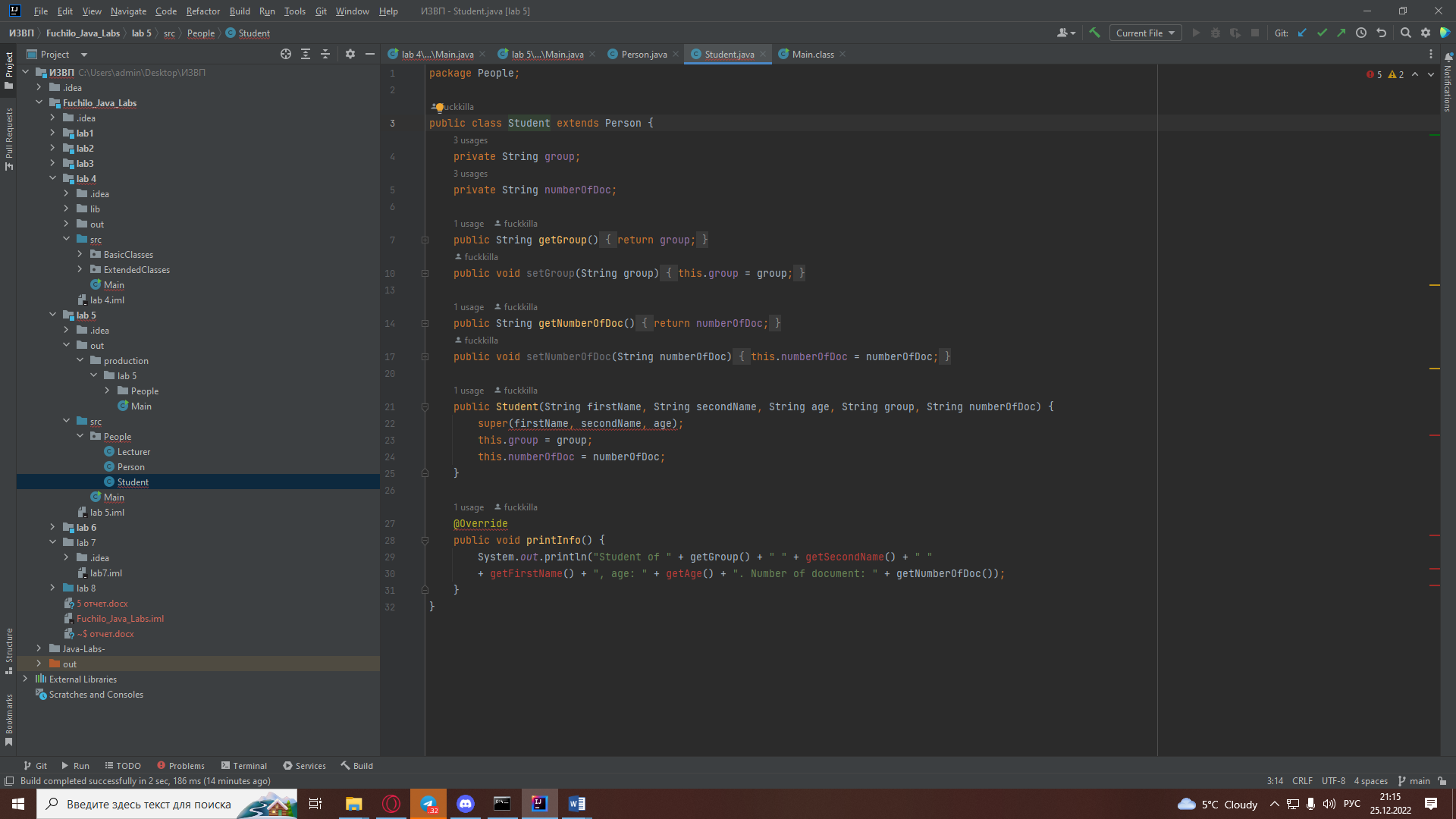


Рисунок 4

3. Создайте класс Lecturer, который должен наследоваться от класса Person. Добавьте дополнительные поля, и геттеры\сеттеры для этих полей:

* Кафедра;
* Зарплата.

Класс Lecturer изображен на рисунке 5

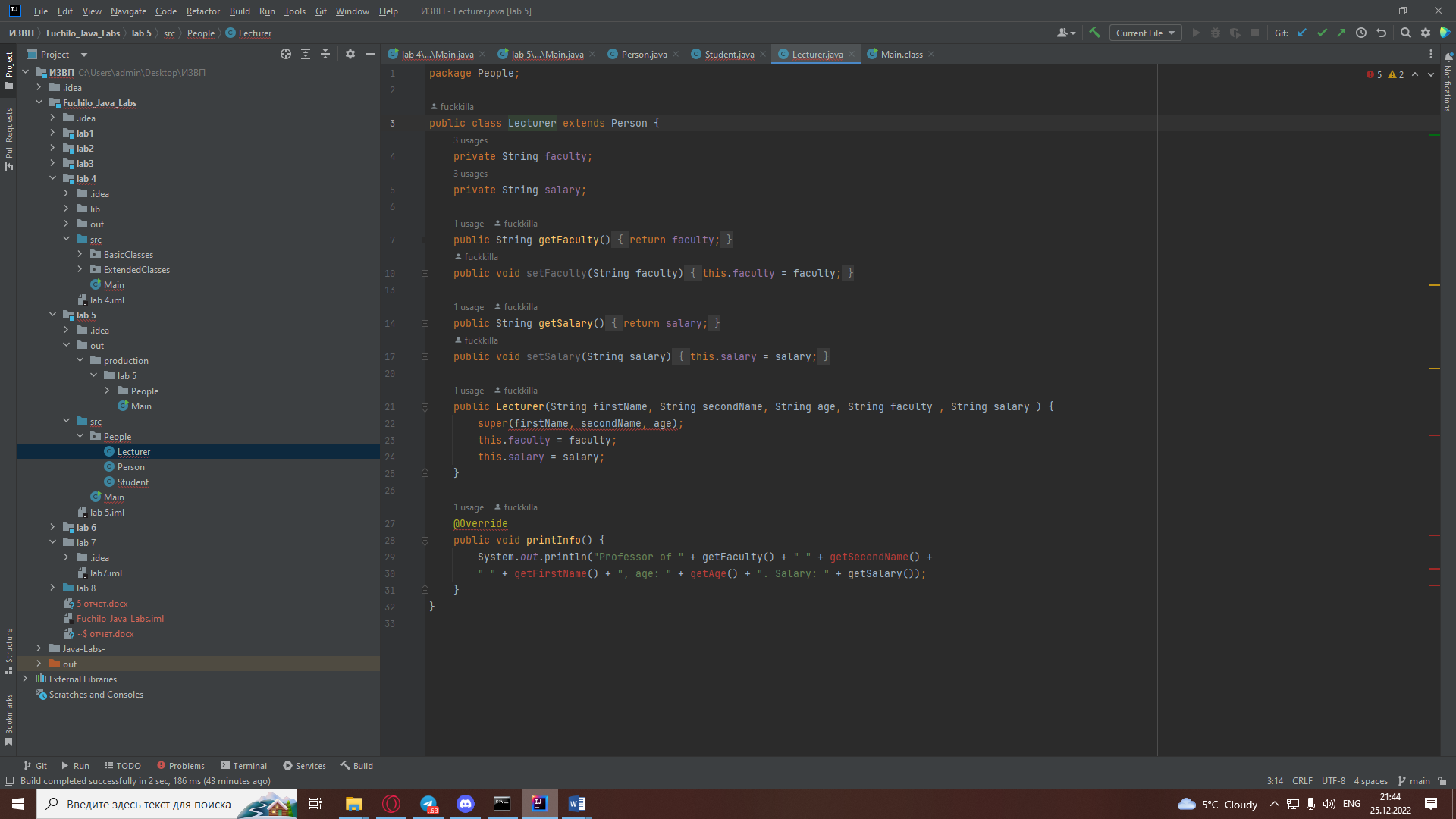


Рисунок 5

Переопределите метод printInfo(), который должен выводить строку следующего формата:

|  |
| --- |
| Преподаватель кафедры <кафедра> <фамилия> <имя>, возраст: <возраст>. Зарплата: <зарплата> |

Метод printInfo() изображен на рисунке 6

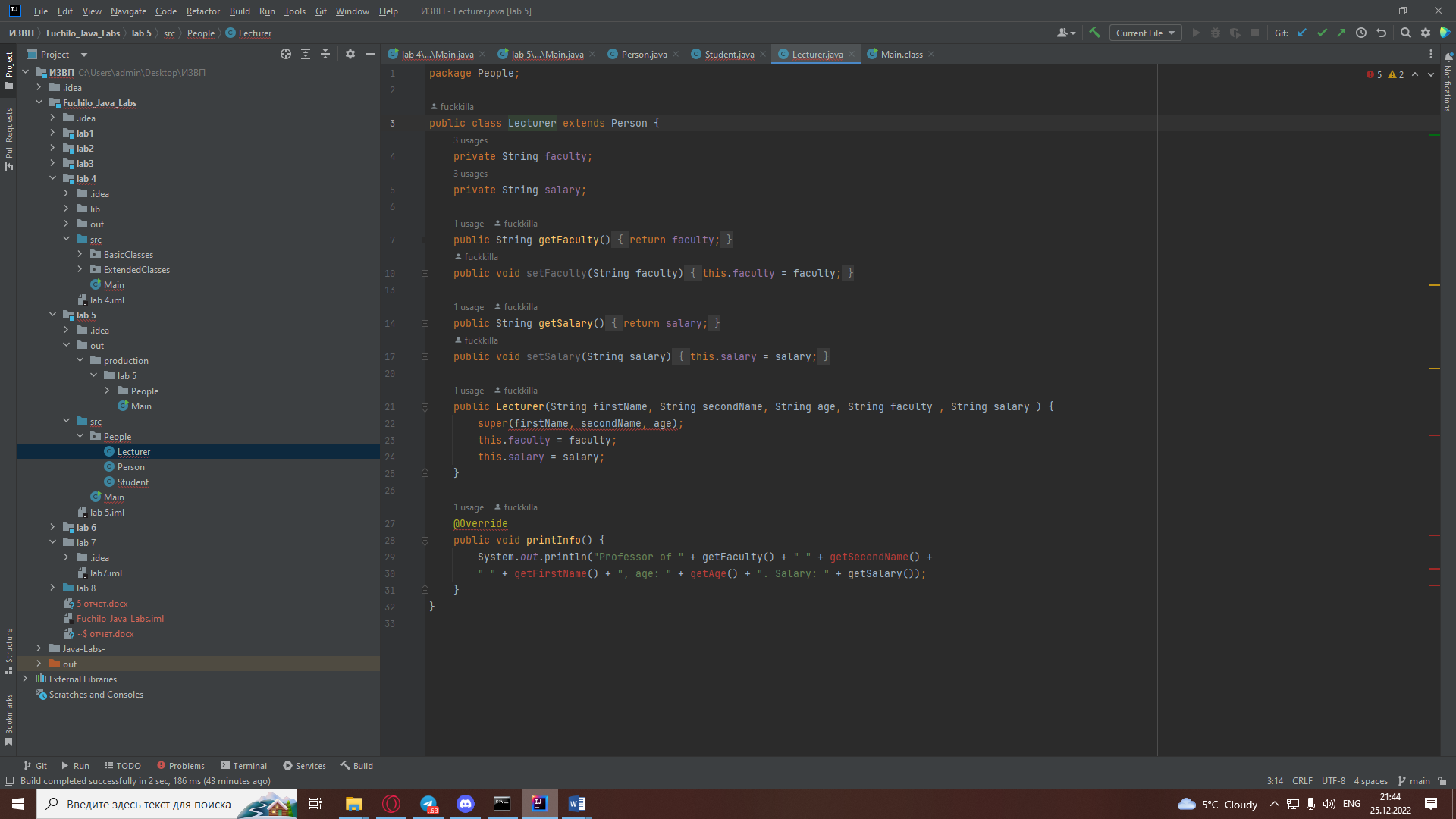


Рисунок 6

4. Используя восходящее преобразование, создайте в классе Main несколько объектов классов Student и Lecturer, после чего создайте массив, который бы мог включать объекты классов Person, Student, Lecturer. Заполните массив объектами этих классов. Изображено на рисунке 7

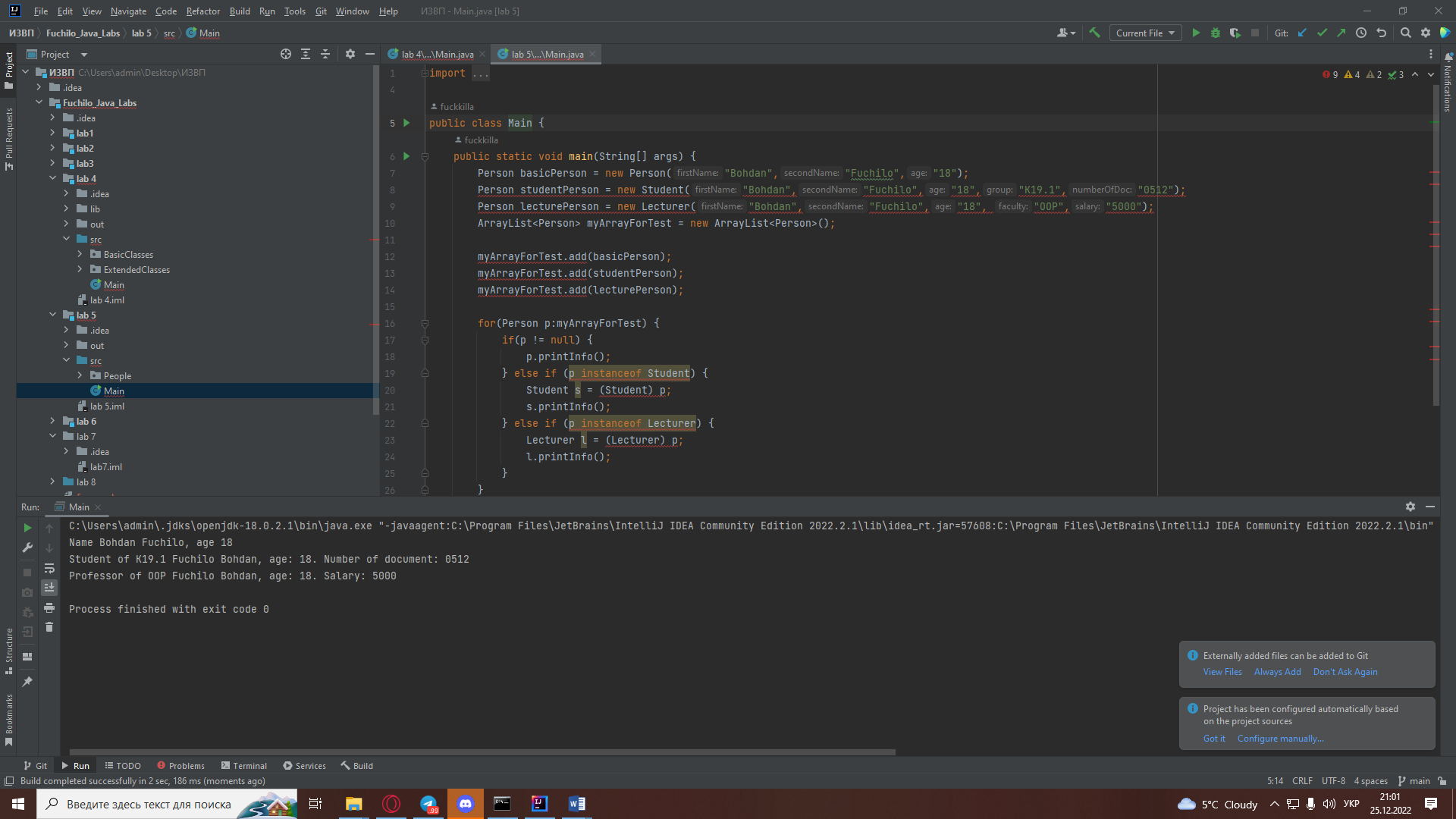


Рисунок 7

5. Используя цикл, обратитесь к элементам массива и выведите в консоль, с помощью метода printInfo(), информацию от каждого объекта. Изображено на рисунке 8

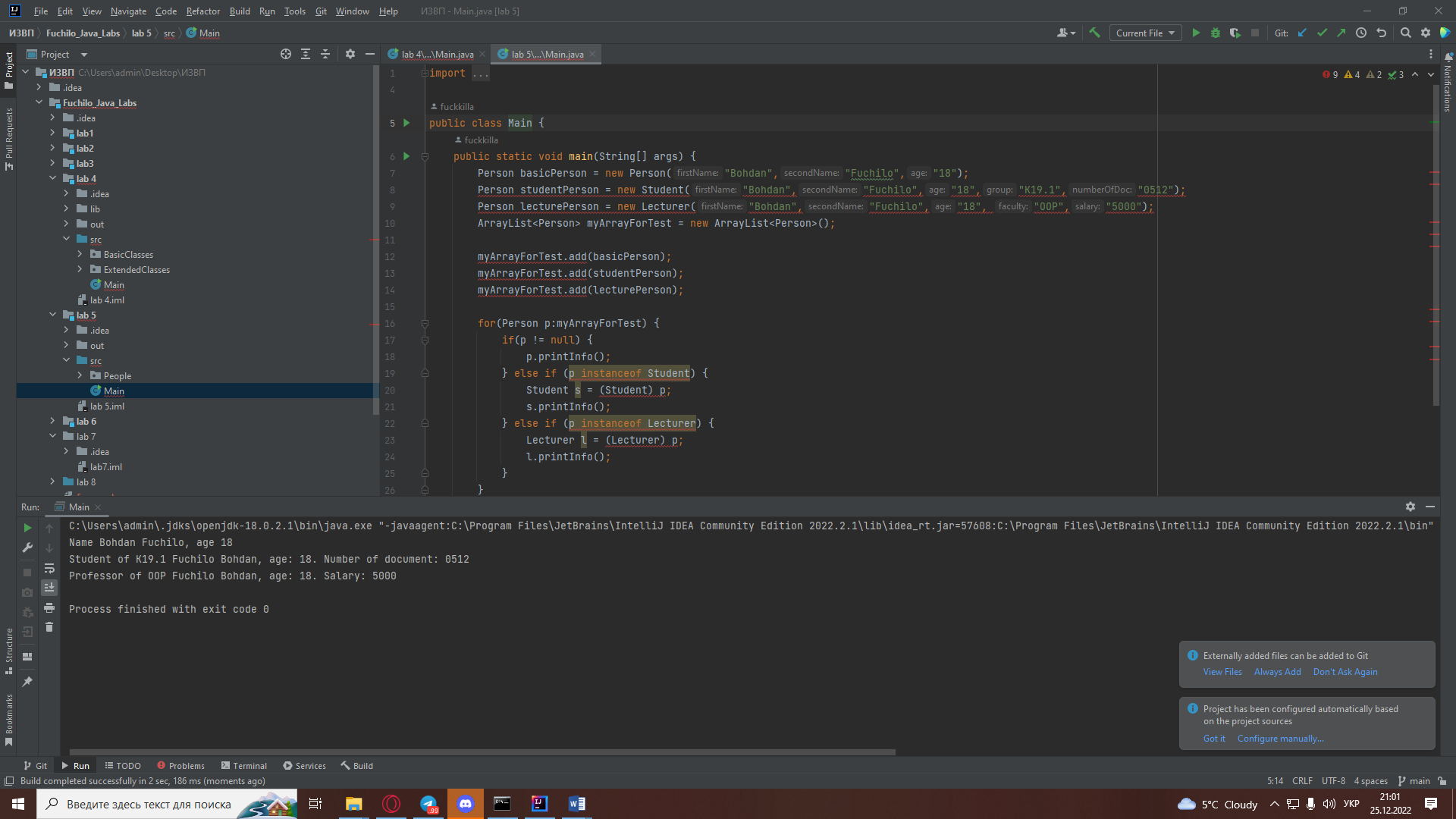


Рисунок 8